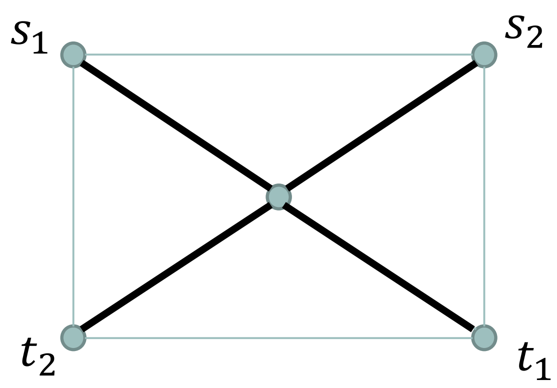
ONLINE DEGREE-BOUNDED STEINER NETWORK DESIGN

Sina Dehghani
Saeed Seddighin
Ali Shafahi
Fall 2015

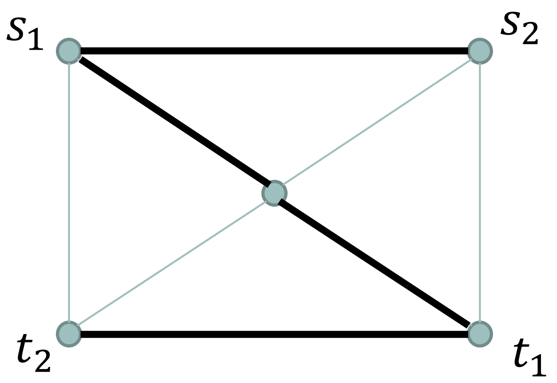
ONLINE STEINER FOREST PROBLEM

- An initially given graph G.
- A sequence of demands (s_i, t_i) arriving one-by-one.
- Buy new edges to connect demands.



DEGREE-BOUNDED STEINER FOREST

- There is a given bound b_v for every vertex v.
- degree violation $:= \frac{deg_H(v)}{b_v}$.
- Find a Steiner forest H minimizing the degree violations.



PREVIOUS OFFLINE WORK

- Degree-bounded network design:

Problem	Paper	Result
Degree-bounded Spanning tree	FR ’90	$O(\log n)$-approximation
Degree-bounded Steiner tree	AKR ’91	$O(\log n)$-approximation
Degree-bounded Steiner forest	FR ’94	maximum degree $\leq b^* + 1$

PREVIOUS OFFLINE WORK

- Edge-weighted degree-bounded variant:

Problem	Paper	Result
EW DB Steiner forest	MRSRRH. '98	$\langle O(\log n), O(\log n)\rangle$-approx.
EW DB Spanning tree	G '06	min weight, max deg $\leq b^* + 2$
EW DB Spanning tree	LS '07	min weight, max deg $\leq b^* + 1$

PREVIOUS ONLINE WORK

- Online weighted Steiner network (no degree bound)

Problem	Paper	Result
Online edge-weighted Steiner tree	IW '91	$O(\log n)$-competitive
Online edge-weighted Steiner forest	AAB '96	$O(\log n)$-competitive

OUR CONTRIBUTION

- Online degree-bounded Steiner network:

Problem	Result
Online degree-bounded Steiner forest	$O(\log n)$-competitive greedy algorithm
Online degree-bounded Steiner tree	$\Omega(\log n)$ lower bound
Online edge-weighted degree-bounded Steiner tree	$\Omega(n)$ lower bound
Online degree-bounded group Steiner tree	$\Omega(n)$ lower bound for det. algorithms.

LINEAR PROGRAM

$\forall e \in E$: $x(e) = 1$ if and only if e is selected.

$\boldsymbol{S}$ be the collection of separating sets of demands.

OMPC has an $O(\log^2 n)$-competitive fractional solution, but rounding that is hard!

$$\min \alpha$$

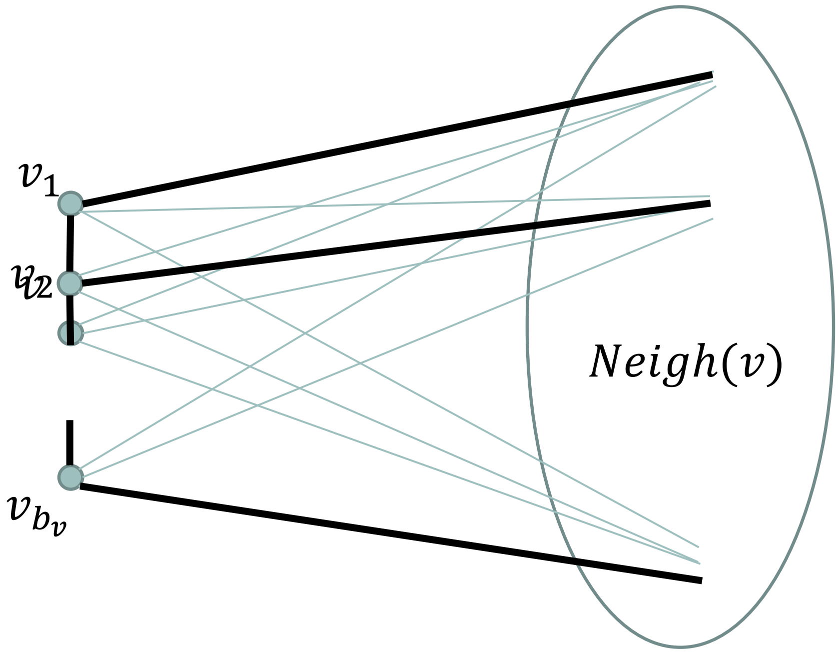
$$\forall v \in V \sum_{e \in \delta(v)} x(e) \leq \alpha . b_v$$ limits degree violations.

$$\forall S \in \boldsymbol{S} \sum_{e \in \delta(S)} x(e) \geq 1$$ ensures connectivity.

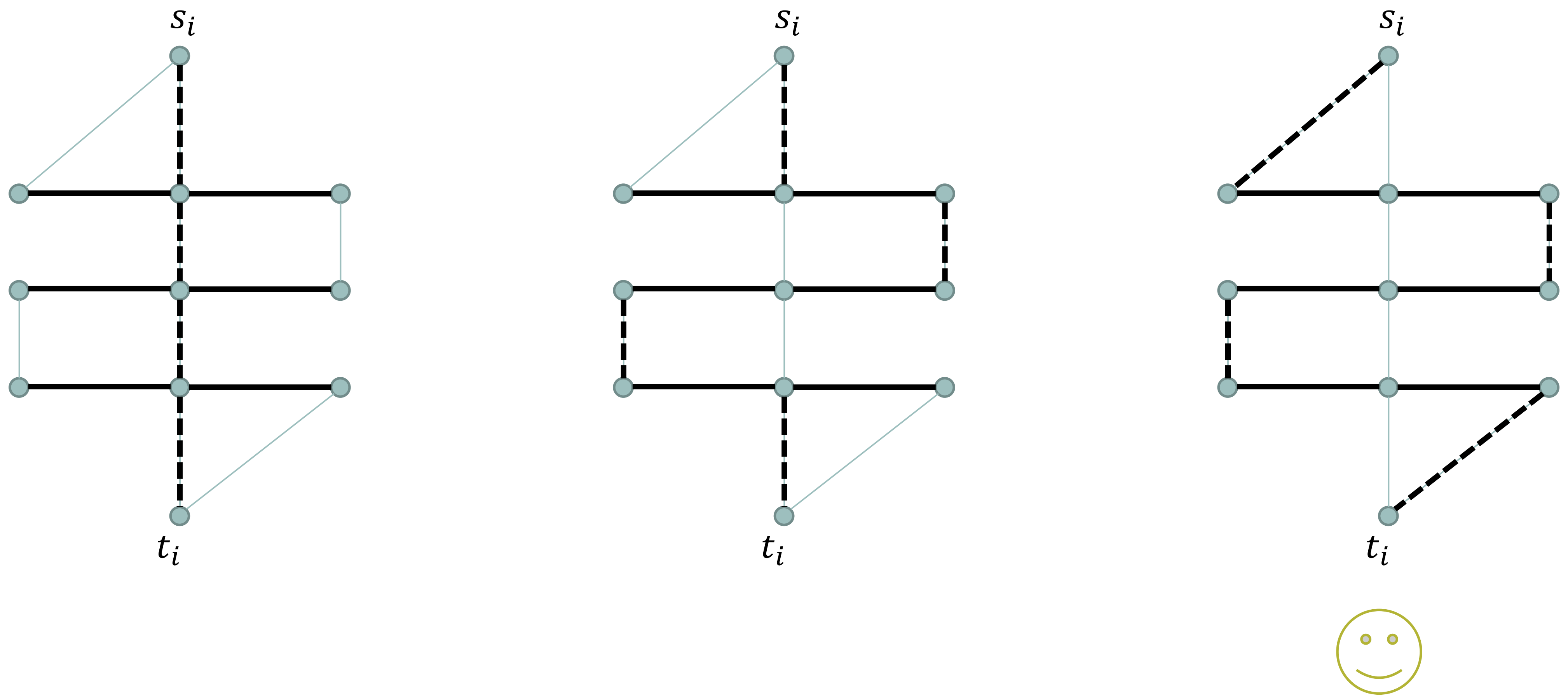
$$\boldsymbol{x}(e), \alpha \in \mathbb{R}^+$$

REDUCTION TO UNIFORM DEGREE BOUNDS

- Replace v with $v_1 \dots v_{b_v}$.
- Connect each v_i to all neighbors of v.
- Set all degree bounds to 1.
- Uniformly distribute edges of $\delta_H(v)$ among v_i's.
- The degree violation remains almost the same.



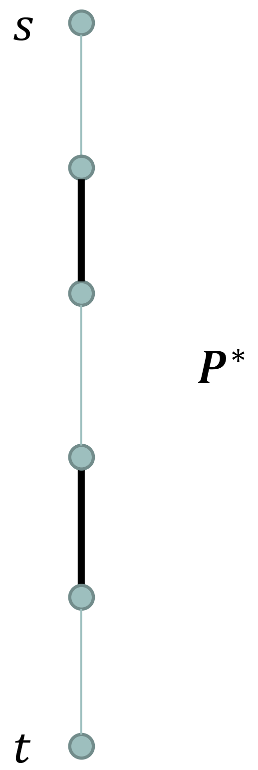
GREEDY ALGORITHM



GREEDY ALGORITHM

- Definitions:
  - Let H denote the online output of the previous step.
  - For an (s, t)-path P the extension part is $\mathrm{P}^* = \{e | e \in P, e \notin H\}$.
  - The load of P^* is $l_H(P^*) = \max_{v \in P^*} \deg_H(v)$.
- Algorithm:
  1. Initiate $H = \phi$.
  2. For every new demand (s_i, t_i):
     1. Find the path P_i with the minimum $l_H(P_i^*)$.
     2. $H = H \cup P_i^*$.

Can be done polynomially.



ANALYSIS

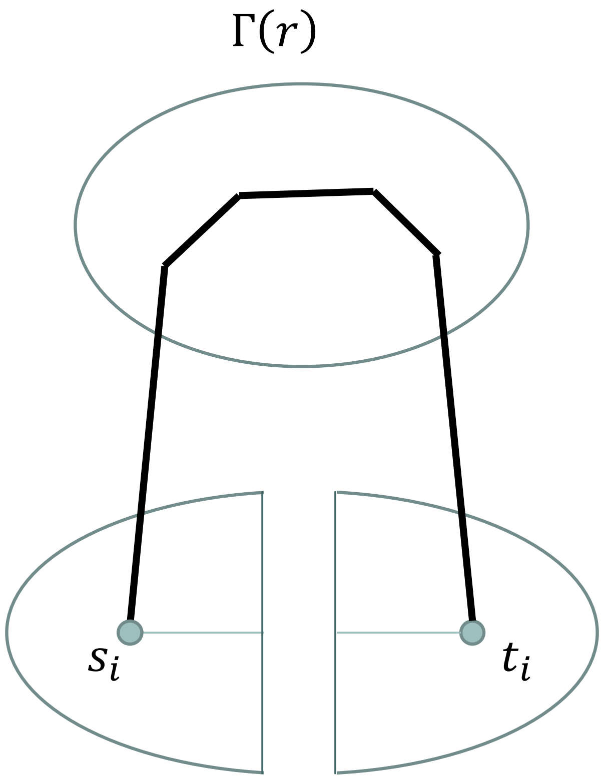
- Let $\Gamma(r)$ be the set of vertices with $\deg_H(v) \geq r$.
- Let $D(r)$ be demands for which $l_H(P_i^*)$ is at least r.

***Remark**: $\Gamma(r)$ is a cut-set for s_i and t_i for every $i \in D(r)$.*

- Let $CC(r)$ denote the number of connected components of $G \setminus \Gamma(r)$ that have at least one endpoint of demand i $\in D(r)$.

Lemma: $\forall r\colon\ CC(r) \geq |D(r)| + 1.$

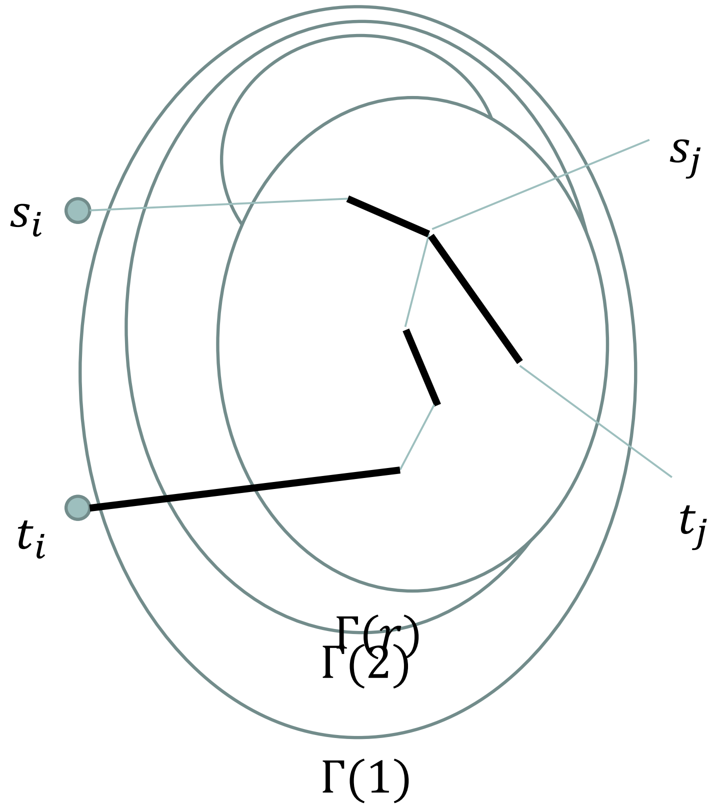
Remark: $\forall \mathrm{r}\colon\ OPT \geq \frac{CC(r)}{|\Gamma(r)|}$.



ANALYSIS

- $\Gamma(r)$'s have a hierarchical order, i.e. $\Gamma(r+1) \subseteq \Gamma(r)$.
- Every demand $i \in D(r)$ copies some vertices to upper level.
- Out of all copies, at most $2(\Gamma(r)-1)$ are for internal edges.

Lemma: $\forall r: |D(r)| \geq \sum_{t=r+1}^{\Delta} |\Gamma(t)| - 2(\Gamma(r)-1)$.



ANALYSIS

Lemma: *For every sequence of integers* $a_1 \geq a_2 \geq \cdots \geq a_\Delta > 0$

$$\max_i \{\frac{\sum_{j=i}^{\Delta} a_j}{a_i}\} \geq \frac{\Delta}{2 \log a_1} .$$

- Partition to $\log a_1$ groups.
- One group has at least $\frac{\Delta}{\log a_1}$ numbers.

$$a_1 \geq a_2 \geq \quad \dots \quad \geq a_i \geq \ \dots \dots \ \dots \geq a_\Delta$$

$2^{\lfloor \log a_1 \rfloor}$ 2^k 2^{k-1} 2^0

ANALYSIS

- Putting all together:

$$\forall r: OPT \geq \frac{CC(r)}{|\Gamma(r)|} \Rightarrow OPT \geq \max_r \frac{CC(r)}{|\Gamma(r)|}.$$

$$CC(r) \geq |D(r)| + 1.$$

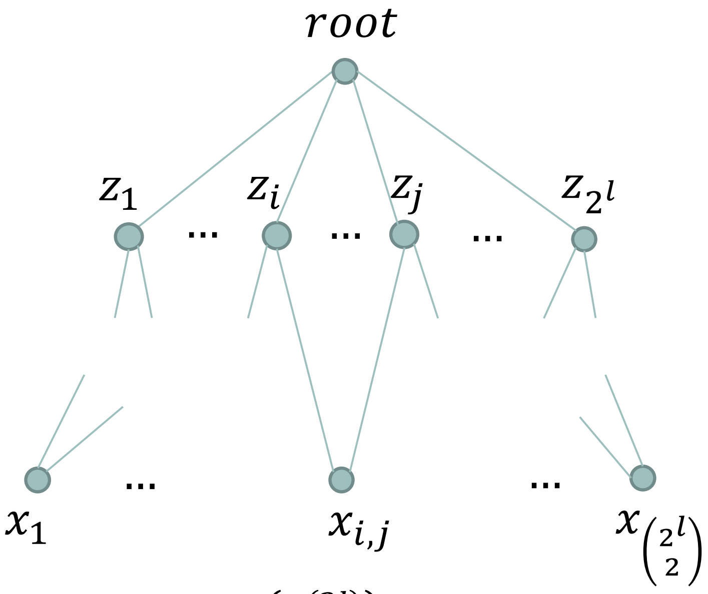
$$D(r) \geq \sum_{t=r+1}^{\Delta} |\Gamma(t)| - 2(\Gamma(r) - 1).$$

- Setting $a_i = |\Gamma(i)|$ and using the lemma:

$$OPT \geq \max_r \frac{\sum_{t=r}^{\Delta} |\Gamma(r)| - O(|\Gamma(r)|) + 1}{|\Gamma(r)|} \geq \frac{\Delta}{2\log|\Gamma(1)|} - O(1) \in \Omega(\frac{\Delta}{\log n})$$

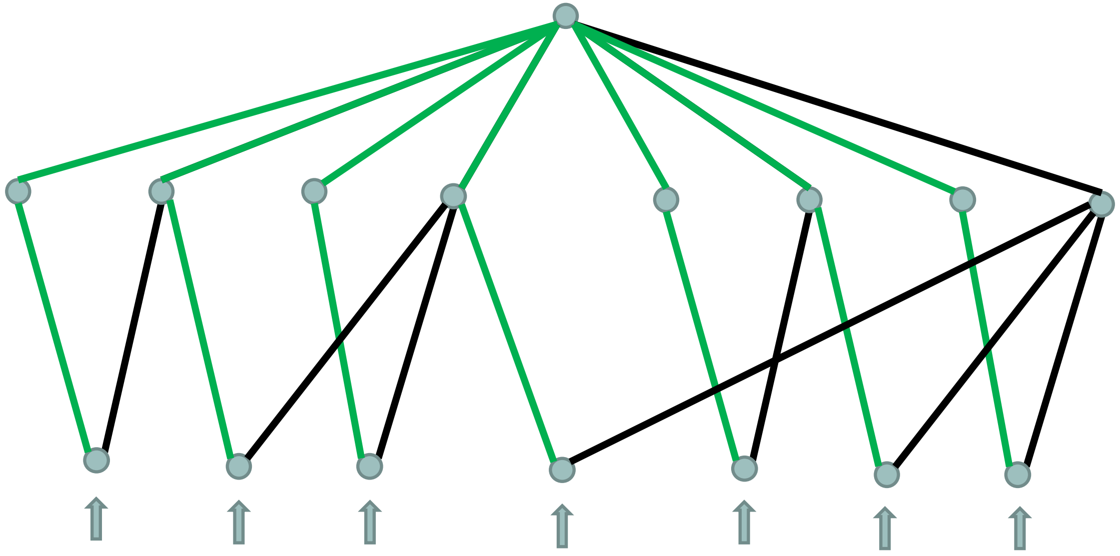
LOWER BOUND

Theorem: *Every (randomized) algorithm for online degree-bounded Steiner tree is* $\Omega(\log n)$-competitive.



$$n \in O\left(2^{(2l)}\right)$$

$$b_v = \begin{cases} n & if\ v = root \\ 2 & O.W. \end{cases}$$



LOWER BOUND

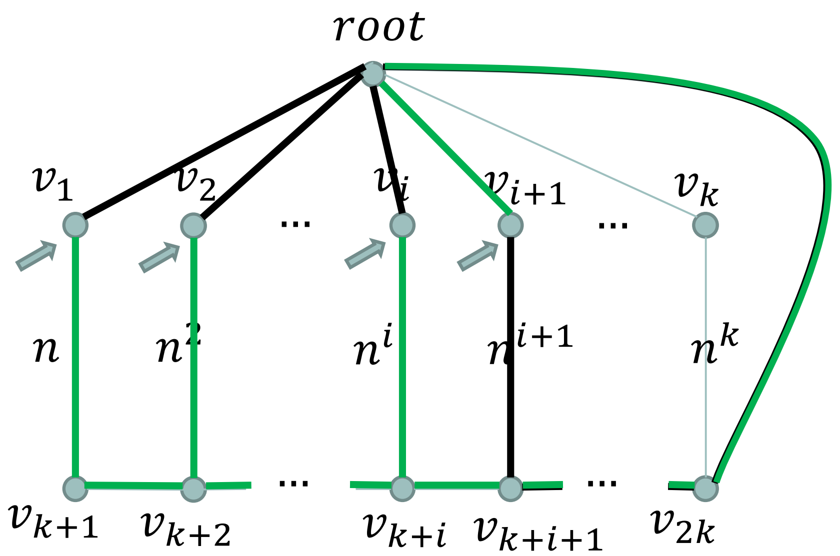
- ***Theorem:*** *Let OPT_b denote the minimum weight of a Steiner tree with maximum degree b. Then for every (randomized) algorithm A for online edge-weighted degree-bounded Steiner tree either*
  - $E[\max \deg_A(v)] \geq \Omega(n).b$

  or
  - $E[weight(A)] \geq \Omega(n).OPT_b.$

$$n = 2k + 1$$
$$b = 3$$

$$weight(A) = n^{i+1}$$
$$\deg_A(root) = i$$

$$OPT_3 = \sum_{j=1}^{i} n^j \in O(n^i)$$

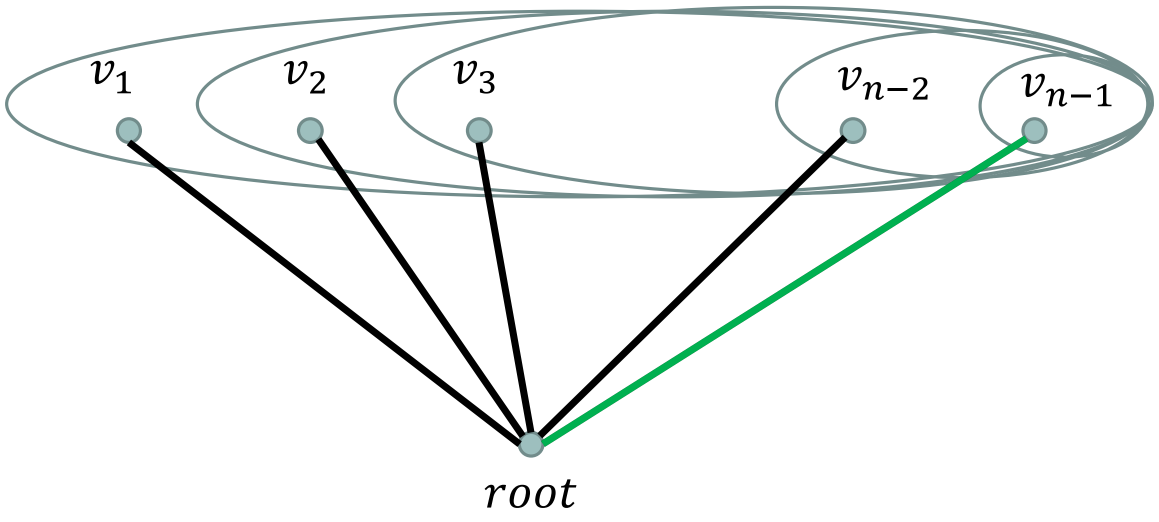


LOWER BOUND

Theorem: *Every deterministic algorithm A for online degree-bounded group Steiner tree is $\Omega(n)$-competitive.*

All degree bounds are 1.

$\deg_A(root) = n - 1.$



OPEN PROBLEMS

- The main open problem:
  - Online edge-weighted degree-bounded Steiner forest,

    when the weights are polynomial to n.
- Other degree-bounded variants (with or without weights):
  - Online group Steiner tree.
  - Online survivable network design.

Thank you